PHASE1-1.4 PART 2: Azure Cost Collector - Execution & Validation

OptiInfra Development Series

Phase: Cost Agent (Week 2-3)

Component: Azure Cost Collector Validation

Estimated Time: 20 minutes setup + 15 minutes validation

Dependencies: PHASE1-1.4 PART 1, PHASE1-1.2 (AWS), PHASE1-1.3 (GCP)

📋 Overview

This document provides validation steps for the Azure Cost Collector to ensure it's collecting accurate cost data from Azure Cost Management API.

✅ Pre-Validation Checklist

- [ ] Azure Cost Collector code generated from PART 1
- [ ] Azure subscription with Cost Management enabled
- [ ] Service principal credentials created
- [ ] At least 2 weeks of Azure usage data
- [ ] Cost Agent service running on port 8001
- [ ] AWS and GCP collectors completed (for 3-cloud comparison)

🔐 Step 0: Azure Service Principal Setup

0.1 Create Service Principal

bash

```bash
```

```
export AZURE_SUBSCRIPTION_ID="your-subscription-id"
export SP_NAME="optiinfra-cost-collector"

# Create service principal
az ad sp create-for-rbac \
  --name $SP_NAME \
  --role "Cost Management Reader" \
  --scopes /subscriptions/$AZURE_SUBSCRIPTION_ID

# Output will show:
# {
#   "appId": "xxxx-xxxx-xxxx-xxxx",
#   "displayName": "optiinfra-cost-collector",
#   "password": "xxxx-xxxx-xxxx-xxxx",
#   "tenant": "xxxx-xxxx-xxxx-xxxx"
# }

# Save credentials
export AZURE_CLIENT_ID="<appId from above>"
export AZURE_CLIENT_SECRET="<password from above>"
export AZURE_TENANT_ID="<tenant from above>"
```

0.2 Grant Required Permissions

```bash
```

```bash
# Cost Management Reader (already assigned)
az role assignment create \
  --assignee $AZURE_CLIENT_ID \
  --role "Cost Management Reader" \
  --scope /subscriptions/$AZURE_SUBSCRIPTION_ID

# Reader (for resource metadata)
az role assignment create \
  --assignee $AZURE_CLIENT_ID \
  --role "Reader" \
  --scope /subscriptions/$AZURE_SUBSCRIPTION_ID

# Monitoring Reader (for metrics)
az role assignment create \
  --assignee $AZURE_CLIENT_ID \
  --role "Monitoring Reader" \
  --scope /subscriptions/$AZURE_SUBSCRIPTION_ID

# Wait for RBAC propagation
sleep 60
```

0.3 Verify Permissions

bash

```bash
# Login as service principal
az login --service-principal \
  -u $AZURE_CLIENT_ID \
  -p $AZURE_CLIENT_SECRET \
  --tenant $AZURE_TENANT_ID

# Test Cost Management access
az consumption usage list --top 1

# Test resource access
az vm list --query "[0].name"
```

0.4 Set Environment Variables

bash

```
export AZURE_SUBSCRIPTION_ID="your-subscription-id"
export AZURE_TENANT_ID="your-tenant-id"
export AZURE_CLIENT_ID="your-client-id"
export AZURE_CLIENT_SECRET="your-client-secret"

# Add to ~/.bashrc for persistence
cat >> ~/.bashrc << EOF
# OptiInfra Azure Configuration
export AZURE_SUBSCRIPTION_ID="your-subscription-id"
export AZURE_TENANT_ID="your-tenant-id"
export AZURE_CLIENT_ID="your-client-id"
export AZURE_CLIENT_SECRET="your-client-secret"
EOF
```

🚀 Step 1: Install Dependencies & Start Service

```bash
cd ~/optiinfra/services/cost-agent

pip install azure-mgmt-costmanagement==4.0.0 \
    azure-mgmt-compute==30.0.0 \
    azure-mgmt-monitor==6.0.0 \
    azure-mgmt-sql==4.0.0 \
    azure-identity==1.14.0

pkill -f "python -m src.main"
python -m src.main
```

Expected Output:

```
INFO:     Starting Cost Agent...
INFO:     AWS Cost Collector initialized
INFO:     GCP Cost Collector initialized
INFO:     Azure Cost Collector initialized
INFO:     Registered collectors: aws, gcp, azure
INFO:     Application startup complete.
```

📊 Step 2: Test Azure Connection

```bash
curl -X POST http://localhost:8001/api/v1/azure/test-connection | jq
```

Expected Response:

```json
{
  "status": "connected",
  "subscription_id": "your-subscription-id",
  "tenant_id": "your-tenant-id",
  "credentials_valid": true,
  "permissions_valid": true,
  "cost_management_accessible": true
}
```

💰 Step 3: Collect Azure Costs

```bash
curl -X POST http://localhost:8001/api/v1/azure/collect \
  -H "Content-Type: application/json" \
  -d '{
    "start_date": "2025-10-01",
    "end_date": "2025-10-31",
    "analyze": true
  }' | jq
```

Expected Response:

```json
{
  "status": "started",
  "job_id": "azure-collect-20251021-120045",
  "estimated_duration_seconds": 20,
  "services_to_collect": 4
}
```

📈 Step 4: Query Cost Data

```bash
curl "http://localhost:8001/api/v1/azure/costs?start_date=2025-10-01&end_date=2025-10-31" | jq
```

Expected Response:

```json
{
  "total_cost": 88000.50,
  "by_service": {
    "Virtual Machines": 58000.00,
    "Azure SQL Database": 16000.00,
    "Azure Functions": 8000.50,
    "Storage": 4500.00
  },
  "by_region": {
    "eastus": 48000.00,
    "westus2": 25000.00,
    "westeurope": 15000.50
  }
}
```

🎯 Step 5: Check Opportunities

```bash
curl "http://localhost:8001/api/v1/azure/opportunities?type=spot_vm" | jq
```

Expected: Spot VM opportunities (70-90% savings)

🌐 Step 6: Multi-Cloud Comparison

```bash
curl "http://localhost:8001/api/v1/multi-cloud/compare?start_date=2025-10-01&end_date=2025-10-31" | jq
```

Expected Response:

```json
{
  "comparison": {
    "aws": {"total_cost": 120000.50},
    "gcp": {"total_cost": 95000.75},
    "azure": {"total_cost": 88000.50}
  },
  "cheapest_provider": "azure",
  "total_cost": 303000.75,
  "total_savings_potential": 142000.00
}
```

💾 Step 7: Verify ClickHouse

```bash
clickhouse-client --query "
SELECT provider, SUM(cost) as total
FROM cost_metrics
WHERE date >= today() - 30
  AND provider IN ('aws', 'gcp', 'azure')
GROUP BY provider
ORDER BY total DESC
"
```

Expected Output:

```
┌─provider─┬─total──────────┐
│ aws      │ 120000.50 │
│ gcp      │  95000.75 │
│ azure    │  88000.50 │
└──────────┴────────────────┘
```

📊 Step 8: Check Prometheus Metrics

```bash
curl http://localhost:8001/metrics | grep azure_total_monthly_cost_usd
```

Expected:

```
azure_total_monthly_cost_usd{service="Virtual Machines"} 58000.0
azure_total_monthly_cost_usd{service="Azure SQL Database"} 16000.0
```

🧪 Step 9: Run Tests

```bash
pytest tests/collectors/test_azure.py -v
# Expected: 16/16 pass

pytest tests/integration/test_azure_integration.py -v
# Expected: 6/6 pass
```

🐛 Troubleshooting